WHAT IS CLAIMED IS:

1. An architecture for a communications node in a telecommunications network, said node performing a plurality of call-control functions using an operating system and a single physical platform, said architecture comprising:

a plurality of application-level logic blocks corresponding to the plurality of call-control functions; and

a common engine module interfaced with the application-level logic blocks, said engine module comprising:

a plurality of functional blocks, selected ones of said functional blocks being operable to perform selected ones of the call-control functions when interfaced with selected ones of the application-level logic blocks; and

at least one mapping table that interfaces the plurality of application-level logic blocks with the plurality of functional blocks in the common engine module, and selects appropriate functional blocks for matching with the application-level logic blocks.

2. The architecture for a communications node of claim 1 wherein the mapping table includes groups of network addresses for application-level logic blocks and for functional blocks in the common engine module, each of said groups of addresses identifying a selected application-level logic block and at least one functional block in the common engine module that together perform the call-control function corresponding to the selected application-level logic block.

3. The architecture for a communications node of claim 2 further comprising:

a plurality of servlet Application Programming Interfaces (APIs) that are operable to provide a plurality of supplementary user services; and

a servlet manager interfaced with the plurality of servlet APIs and with the plurality of application-level logic blocks, said manager being operable to provide selected ones of the supplementary user services to any one of the application-level logic blocks.

4. The architecture for a communications node of claim 1 wherein the telecommunications network utilizes call-control signaling based on the Session Initiation Protocol (SIP), and the plurality of application-level logic blocks include logic blocks for a Call State Control Function (CSCF).

5. The architecture for a communications node of claim 4 wherein the plurality of functional blocks in the common engine module include a plurality of SIP behavior functions and a SIP stack that performs reliability and error-checking functions associated with signal communications with the communications node.

6. The architecture for a communications node of claim 5 wherein the plurality of SIP behavior functions includes a proxy function, a User Agent Server (UAS) function, and a User Agent Client (UAC) function.

7. The architecture for a communications node of claim 5 wherein at least one of the application-level logic blocks includes a Registrar SIP behavior function.

8. The architecture for a communications node of claim 5 wherein the SIP stack includes a plurality of portable units, said portable units including:

a transaction manager;

a parser; and

a utility package.

9. An architecture for a Call State Control Function (CSCF) node in a Session Initiation Protocol (SIP) telecommunications network, said node performing call-control functions of a Proxy CSCF (P-CSCF), an Interrogating CSCF (I-CSCF), and a Serving CSCF (S-CSCF), said architecture being implemented on top of a single operating system and a single physical platform, said architecture comprising:

an application-level logic block corresponding to the P-CSCF;

an application-level logic block corresponding to the I-CSCF;

an application-level logic block corresponding to the S-CSCF; and

a common engine module interfaced with the application-level logic blocks, said engine module comprising:

a plurality of SIP behavior functions and a plurality of SIP stack functions, selected SIP behavior functions and selected SIP stack functions being operable to perform the functions of a P-CSCF, I-CSCF, or S-CSCF when interfaced with an appropriate application-level logic block corresponding to the P-CSCF, I-CSCF, or S-CSCF; and

at least one mapping table that interfaces the plurality of application-level logic blocks with the plurality of SIP behavior functions and the SIP stack, and selects appropriate SIP behavior functions and SIP stack functions for matching with the application-level logic blocks.

10. The architecture for a CSCF node of claim 9 further comprising:

a plurality of servlet Application Programming Interfaces (APIs) that are operable to provide a plurality of supplementary user services; and

a servlet manager interfaced with the plurality of servlet APIs and with the application-level logic blocks, said manager being operable to provide selected ones of the supplementary user services to any one of the application-level logic blocks.

11. The architecture for a CSCF node of claim 9 wherein the plurality of SIP behavior functions in the common engine module includes a proxy function, a User Agent Server (UAS) function, and a User Agent Client (UAC) function.

12. The architecture for a CSCF node of claim 9 wherein at least one of the application-level logic blocks includes a Registrar SIP behavior function.

13. A method of implementing a communications node in a telecommunications network, said node performing a plurality of Session Initiation Protocol (SIP) call-control functions using a single operating system and a single physical platform, said method comprising the steps of:

providing a plurality of application-level logic blocks corresponding to the plurality of call-control functions;

assigning a network logic-block address to each of the application-level logic blocks;

interfacing with the application-level logic blocks, a common engine module comprising a mapping table, a plurality of SIP stack functions, and a plurality of SIP call-control behavior functions;

assigning a network address to each of the SIP stack functions and call-control behavior functions;

implementing the application-level logic blocks, and the common engine module on top of the single operating system and the single physical platform;

storing in the mapping table, the logic-block addresses, SIP stack function addresses, and behavior-function addresses; and

identifying in the mapping table, a plurality of interface groups, each interface group comprising a set of addresses associated with one selected application-level logic block and at least one of the SIP stack functions and call-control behavior functions that, together, perform the call-control function corresponding to the selected application-level logic block.

14. The method of implementing a communications node of claim 13 further comprising the steps of:

providing a plurality of servlet Application Programming Interfaces (APIs) that are operable to provide a plurality of supplemental user services; and

interfacing a servlet manager with the plurality of servlet APIs and with the application-level logic blocks, said manager being operable to provide selected ones of the supplemental user services to any one of the application-level logic blocks.

15. The method of implementing a communications node of claim 13 wherein the plurality of SIP call-control behavior functions in the common engine module includes a proxy function, a User Agent Server (UAS) function, and a User Agent Client (UAC) function.

16. The method of implementing a communications node of claim 15 further comprising the step of implementing a SIP Registrar behavior function in at least one of the application-level logic blocks.

17. An architecture for a communications node in a Session Initiation Protocol (SIP) telecommunications network, said node performing a plurality of call-control functions using a common operating system and being implemented on a single physical platform, said architecture comprising:

means for performing application-level logic corresponding to the plurality of call-control functions;

means for interfacing a plurality of SIP functional blocks with the application-level logic blocks, selected ones of said SIP functional blocks being operable to perform selected ones of the call-control functions when interfaced with selected ones of the application-level logic blocks; and

means for mapping into groups, the plurality of application-level logic blocks and the plurality of SIP functional blocks, each of said groups defining a different one of the plurality of call-control functions performed by the node.